Data Exploration

```
> summary(train)
       v1                  v2              v3              v4
 Min.   :0.0002389   Min.   :1.000   Min.   :30.09   Min.   :1.000
 1st Qu.:0.2294261   1st Qu.:1.062   1st Qu.:35.10   1st Qu.:1.079
 Median :0.4803411   Median :1.134   Median :40.09   Median :1.158
 Mean   :0.4882555   Mean   :1.136   Mean   :40.08   Mean   :1.155
 3rd Qu.:0.7439418   3rd Qu.:1.207   3rd Qu.:45.29   3rd Qu.:1.234
 Max.   :0.9984908   Max.   :1.284   Max.   :49.98   Max.   :1.300
       v5              v6                  v7               v8
 Min.   :2.999   Min.   :0.0000597   Min.   :-3.0328   Min.   :1.483
 1st Qu.:3.454   1st Qu.:0.0241761   1st Qu.:-2.5380   1st Qu.:1.496
 Median :3.963   Median :0.0504044   Median :-2.0406   Median :1.500
 Mean   :3.976   Mean   :0.0497798   Mean   :-2.0242   Mean   :1.500
 3rd Qu.:4.486   3rd Qu.:0.0746565   3rd Qu.:-1.5148   3rd Qu.:1.504
 Max.   :5.000   Max.   :0.0998852   Max.   :-0.9635   Max.   :1.516
       v9                Y
 Min.   :-5.5649   Min.   : 464.5
 1st Qu.:-0.4858   1st Qu.: 629.8
 Median : 0.5061   Median : 820.4
 Mean   : 0.5013   Mean   : 836.2
 3rd Qu.: 1.4764   3rd Qu.:1044.4
 Max.   : 5.5063   Max.   :1269.7
```

Given the summary of the training data, it's necessary to preprocess the data since it present observations at different scales. We perform standardisation to get mean 0 and std 1.

```
# pre-processing so each predictor contribute equally
train_standardize <- as.data.frame(scale(train[1:10]))
train_standardize["Y"] <- list(train$Y)
```

For the first item we are going to check the assumption of linear regression model:

- Linearity
- Independence
- Normality
- Homoscedasticity

We will follow forward selection

Parametric Approach, Linear regression

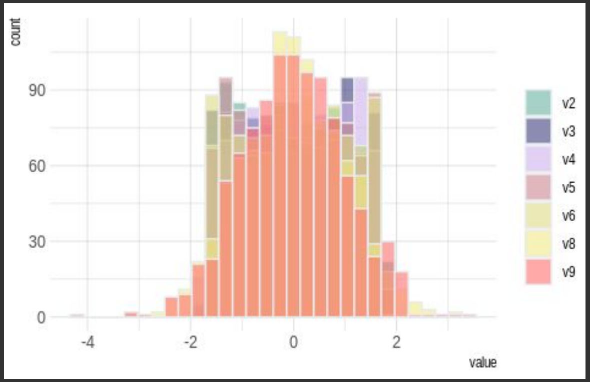
Independence - Correlation

```
# checking correlation between predictors
corr <- cor(train[1:9])# just predictors
which( corr>0.1 & corr<1, arr.ind=TRUE)
```

```
   row col
v5   5   1
v7   7   1
v1   1   5
v7   7   5
v1   1   7
v5   5   7
```

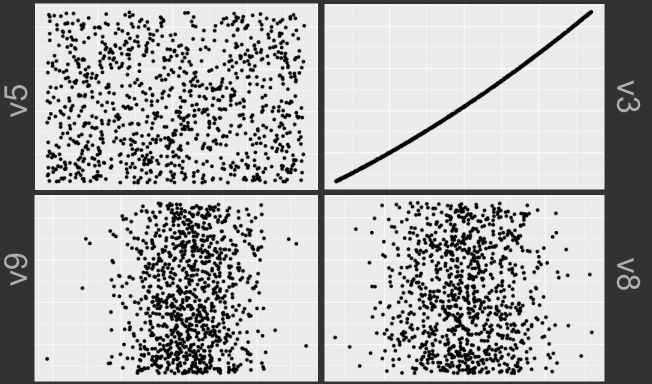
From the correlation analysis v1, v7 and v5 are correlated.
We will keep the predictor with significant p-value ($p < 0.05$).
The dropped predictors are:
V1 and v7

Normality



From the plot we can see some predictors do not follow a normal distribution.
We will remove the predictors with non-significant p-value and that do not fulfil the condition above.
v3 do not follow it but have $p < 0.001$
The dropped predictors are:
v2, v4 and v6

Linearity

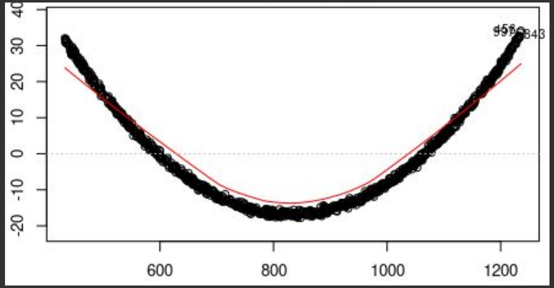


From the plot we can see some predictors can not be described with a straight line.
We will remove the predictors with non-significant p-value that do not fulfil the condition above
v5 do not fulfil but have $p < 0.05$
The dropped predictors are:
None

Parametric Approach, Linear regression

Homoscedasticity variance of error terms

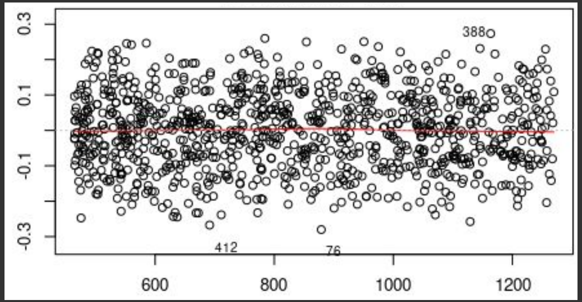
We try the model with the remaining predictors and plot Residual (y) vs Fitted values (x)



In the plot there is no visible randomness. A non-linear model would better describe the data. The relationship can be model as a polynomial regression

Polynomial regression

The residual describe a quadratic behavior. We try the 3 predictors as a quadratic term. The one with significant p-value ($p < 0.001$) is v3



Now there is more randomness in the residuals vs fitted, so the model fit more the data. The predictors fulfil the assumptions of the model

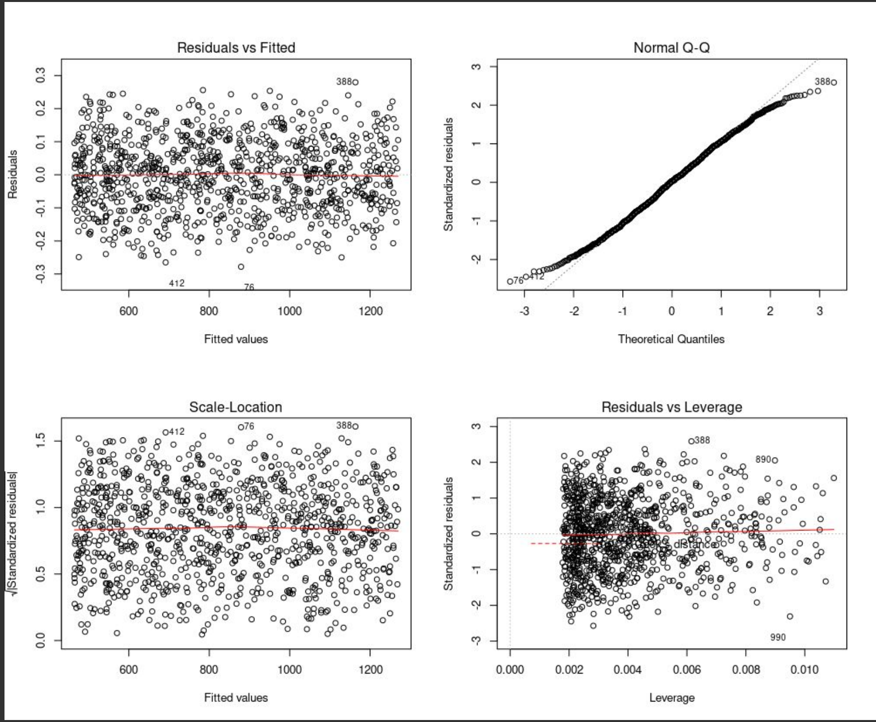
p-value significance

```
> linear_model <- lm(Y ~ v3 + I(v3^2) + v5 + v8 + v9, data=train_standardize)
> summary(linear_model)

Call:
lm(formula = Y ~ v3 + I(v3^2) + v5 + v8 + v9, data = train_standardize)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.280204 -0.078134  0.001416  0.080392  0.273492 

Coefficients:
              Estimate Std. Error    t value Pr(>|t|)    
(Intercept) 819.334864   0.005193 157782.571   <2e-16 ***
v3          236.142109   0.003435  68737.139   <2e-16 ***
I(v3^2)      16.929288   0.003906   4333.674   <2e-16 ***
v5            1.187510   0.003430    346.231   <2e-16 ***
v8           -0.001020   0.003436     -0.297    0.767    
v9            0.004983   0.003438      1.449    0.148    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1083 on 994 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1 
F-statistic: 9.477e+08 on 5 and 994 DF,  p-value: < 2.2e-16
```

From the summary, v8 and v9 p-value ($p > 0.05$) are non-significant. Then we remove them. This predictors have no relationship with the prediction

Linear regression, Results

After the previous analysis the final linear regression model is:




```
> summary(linear_model)

Call:
lm(formula = Y ~ v3 + I(v3^2) + v5, data = train_standardize)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.278145 -0.077925  0.002741  0.078629  0.279523 

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)    
(Intercept) 8.193e+02  5.192e-03 157813.5   <2e-16 ***
v3          2.361e+02  3.432e-03  68799.6   <2e-16 ***
I(v3^2)     1.693e+01  3.905e-03   4335.6   <2e-16 ***
v5          1.187e+00  3.429e-03    346.3   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1083 on 996 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1 
F-statistic: 1.579e+09 on 3 and 996 DF,  p-value: < 2.2e-16
```

Note: Potential problems like outliers, high-leverage points were not present as eliminating v8 and v9 remove the need of the evaluation.

Non-parametric Approach, K-Nearest Neighbors

For the second item by definition no assumptions are made over the model. So we:

- Work in the curse of dimensionality by doing the combinations of predictors and evaluating the relevance of them.
- Find the value of k which reduce the MSE (mean square error)
- Do cross-validation with leave-one-out and majority voting, to find the best fitting model

This results were saved in the variable compare, the predictors, k and MSE values

```r
# for each combination of predictors do the knn for values of k between 1 and sqrt(n)/2 = 15
# also as no test set is given cross validations leave-one-out of the
# training set is perform
for (combination in combinations_list){
  for (i in 1:length(k_values)){
    combination <- combination [! combination %in% zero]
    combination <- unlist(combination, use.names = FALSE)
    # run the knn
    knn_pred = knn.reg(train = train_standard[, combination], test = NULL, y = train$Y, k=k_values[i])
    # save the mse
    row <- data.frame(paste(unlist(combination), collapse=''), k_values[i], mean((knn_pred$residuals)^2))
    names(row) <- c("vars", "k", "mse")
    compare <- rbind(compare, row)
  }
}
```

```
> head(compare)
  vars k       mse
1    1 1 111516.83
2    1 2  86231.42
3    1 3  75428.60
4    1 4  70190.65
5    1 5  67718.01
6    1 6  66331.85
> compare[which(compare$mse == min(compare$mse)), ]
   vars k      mse
50    3 5 2.155141
```

The configuration with lowest MSE is using the v3 predictor and k=5

Conclusions

Linear regression

- Check the assumptions of the model.
- Be aware of the interpretation of summary tables and plots to realize potential problems, information loss.
- At the end 2 out of the 9 predictors where taken as having a relation with the label with a polynomial regression model.

K-Nearest neighbors

- Trying to avoid the curse of dimensionality just one predictor give the best MSE. However the linear regression model conclude 2 predictors. This may be a situation of overfitting of the model.

Without the labels of the test set this cannot be verified.